Name – Jadhav Laxman
Class – SYA
Roll no – 138
Subject : Design & Analysis of Algorithm (DAA)

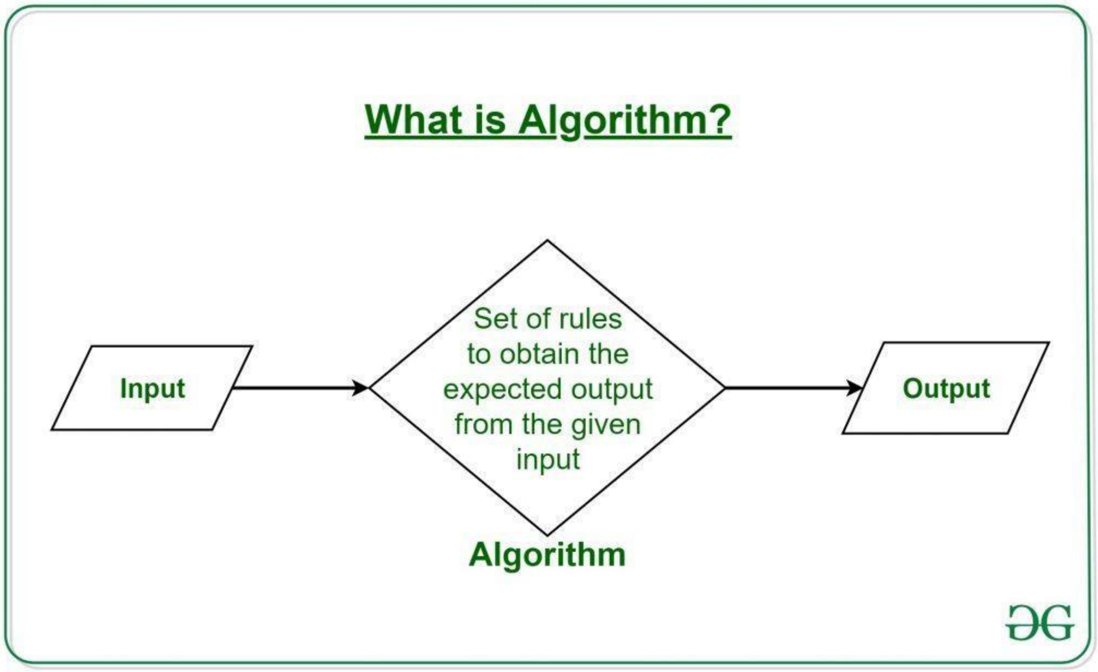
Assignment 1

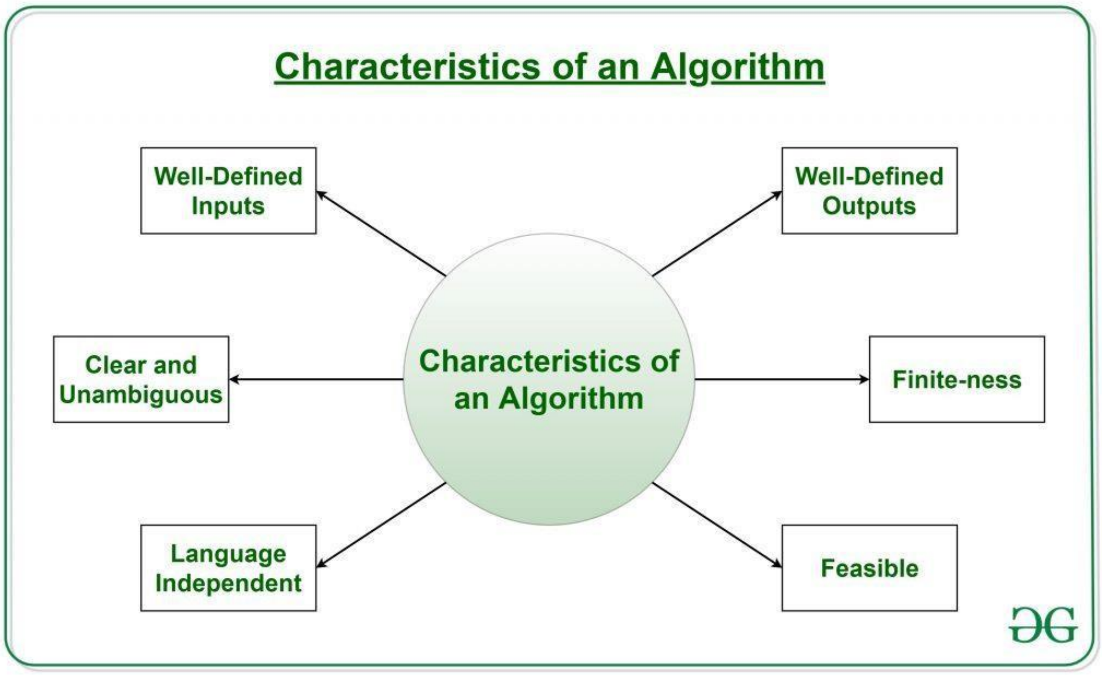
(Design and Analysis of Algorithms)

1. What is an algorithm? Explain different ways of representing an algorithm.

algorithms are simply a series of instructions that are followed, step by step, to do something useful or solve a problem

The word Algorithm means “a process or set of rules to be followed in calculations or other problem-solving operations”. Therefore, Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed upon in order to get the expected results.

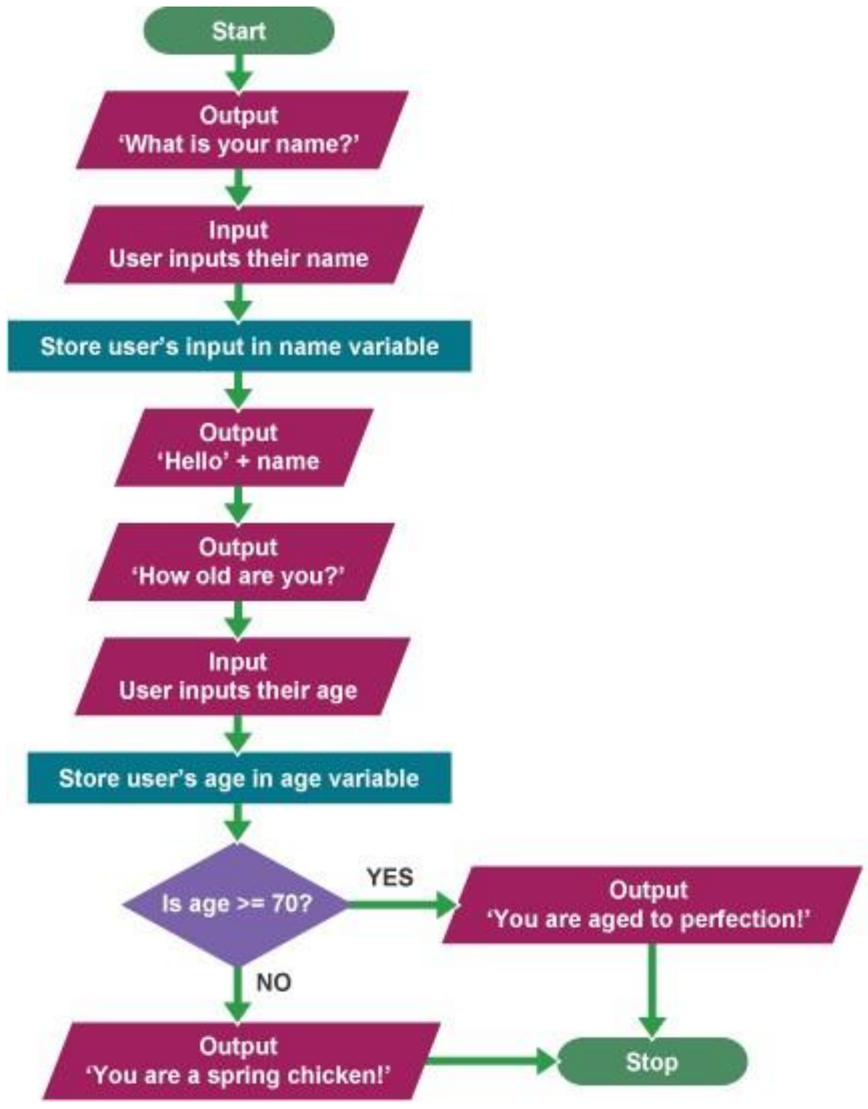

What is Algorithm?
Input
Set of rules to obtain the expected output from the given input
Algorithm
Output



We can express an algorithm many ways, including natural language, flow charts, pseudocode, and of course, actual programming languages.

Flow charts

A flowchart is a diagram that represents a set of instructions. Flowcharts normally use standard symbols to represent the different instructions. There are few real rules about the level of detail needed in a flowchart.



Pseudocode

Pseudocode is not a programming language; it is a simple way of describing a set of instructions that does not have to use specific syntax.

Writing in pseudocode is similar to writing in a programming language. Each step of the algorithm is written on a line of its own in sequence. Usually, instructions are written in uppercase, variables in lowercase and messages in sentence case.

In pseudocode, INPUT asks a question. OUTPUT prints a message on screen.

A simple program could be created to ask someone their name and age, and to make a comment based on these. This program represented in pseudocode would look like this:

```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
    OUTPUT 'You are aged to perfection!'
ELSE
    OUTPUT 'You are a spring chicken!'
```

2. Mention different strategies for writing an algorithm. Explain each strategy in 1 line and specify sample problems which can be solved using each of the strategy.

Selecting a proper design strategy for algorithms is a complex but important task. Following is some of the main algorithm design strategies:

1.Brute-force or exhaustive search

straightforward methods of solving a problem that rely on sheer computing power and trying every possibility rather than advanced techniques to improve efficiency.

- **Example:-** imagine you have a small padlock with 4 digits, each from 0-9. You forgot your combination, but you don't want to buy another padlock. Since you can't remember any of the digits, you have to use a brute force method to open the lock.

2.Divide and Conquer

Divide and Conquer is an algorithmic paradigm. A typical Divide and Conquer algorithm solves a problem using following three steps.

Divide: Break the given problem into subproblems of same type.
Conquer: Recursively solve these subproblems
Combine: Appropriately combine the answers

Example - • Binary Search.
- Quick Sort.
- Merge Sort.

3.Greedy Algorithms

The algorithm makes the optimal choice at each step as it attempts to find the overall optimal way to solve the entire problem.

Examples

- Kruskal's algorithm and Prim's algorithm for finding minimum spanning trees,

- algorithm for finding optimum Huffman trees.

4.Dynamic Programming

Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler subproblems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its subproblems.

Example: - Insertion sort

5.Branch and Bound Algorithm

Branch and bound algorithms are used to find the optimal solution for combinatory, discrete, and general mathematical optimization problems.

6.Randomized Algorithm

An algorithm that uses random numbers to decide what to do next anywhere in its logic is called a Randomized Algorithm.

Example: - in Randomized Quick Sort, we use random number to pick the next pivot (or we randomly shuffle the array)

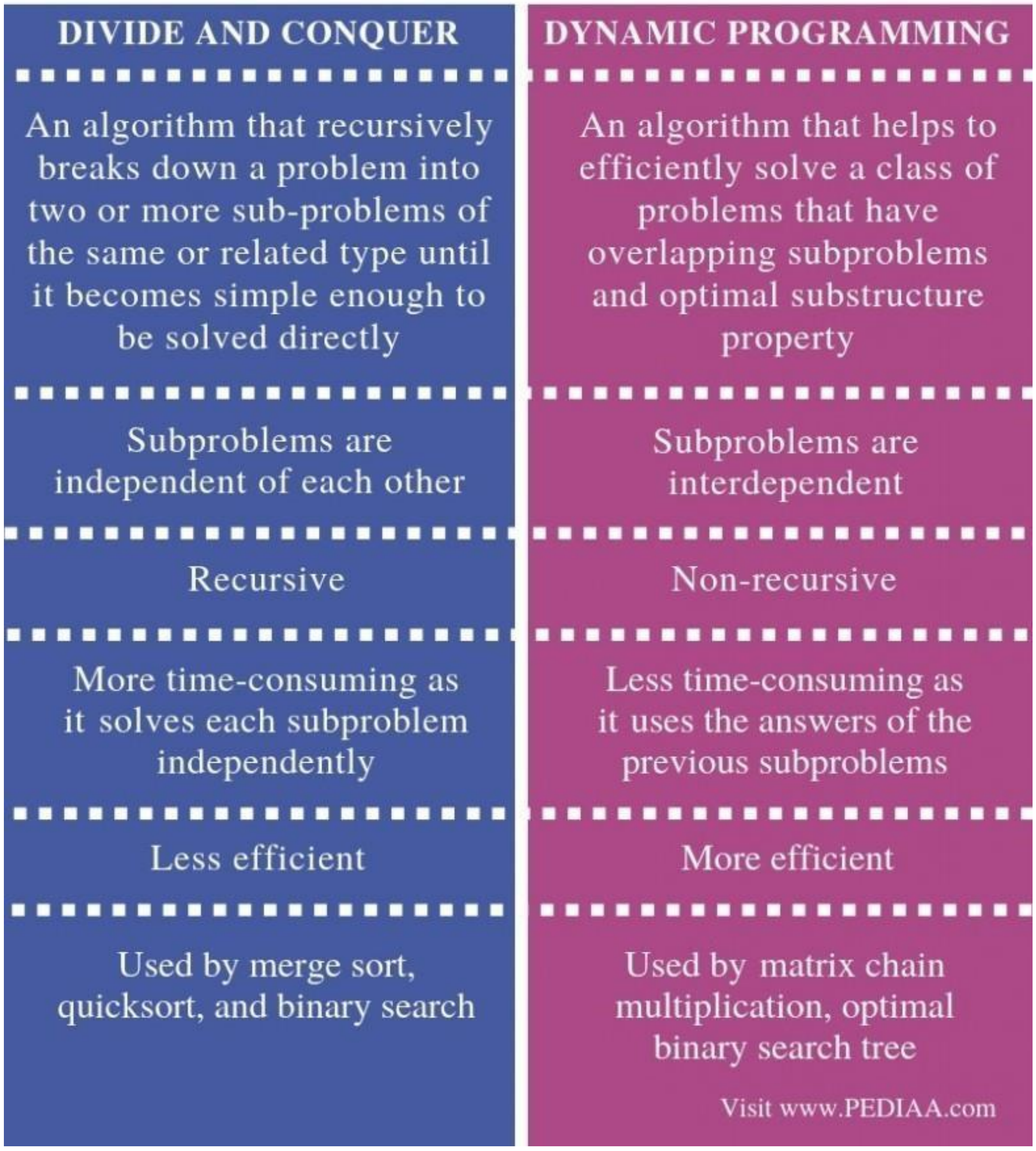
7.Backtracking

Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time

Examples: -

- Puzzle problems such as eight queens puzzle, crosswords, verbal arithmetic, Sudoku, and Peg Solitaire.
- Combinatorial optimization problems such as parsing and the knapsack problem.

3. Differentiate between Divide & Conquer and Dynamic Programming



DIVIDE AND CONQUER	DYNAMIC PROGRAMMING
An algorithm that recursively breaks down a problem into two or more sub-problems of the same or related type until it becomes simple enough to be solved directly	An algorithm that helps to efficiently solve a class of problems that have overlapping subproblems and optimal substructure property
Subproblems are independent of each other	Subproblems are interdependent
Recursive	Non-recursive
More time-consuming as it solves each subproblem independently	Less time-consuming as it uses the answers of the previous subproblems
Less efficient	More efficient
Used by merge sort, quicksort, and binary search	Used by matrix chain multiplication, optimal binary search tree

Visit www.PEDIAA.com